

# NativeSPServiceLogout

The `<Logout>` element is used to enable and configure support for Logout protocols and behavior within the SP. Logout in general can be enabled or disabled by adding or removing it. It replaces the functions of the `<LogoutInitiator>` and `<md:SingleLogoutService>` handler elements from the older (pre-2.4) configuration.

Instead of defining explicit endpoints with low-level binding information, the `<Logout>` element automates the installation of the appropriate handlers based on the protocols selected for activation. Most of the remaining settings are equivalent to the settings supported by the various `<LogoutInitiator>` types.

The use of the `<Logout>` element results in a basic chain of initiator plugins installed at the recommended `"/Logout"` handler location. For advanced scenarios that require additional plugins or options, additional explicit `<LogoutInitiator>` elements can be added to the end of the surrounding `<Sessions>` element.

A basic example supporting SAML 2.0 and "localized" logout:

```
<Logout>SAML2 Local</Logout>
```

## Attributes

- `relayState` (string)
  - Overrides `relayState` setting from the `<Sessions>` element.

Other attributes supported include settings specific to various types of `<LogoutInitiator>` plugins to alter the behavior of specific protocols.

## Element Content

The content of the element is a whitespace-delimited list of "protocol" identifiers. The following are built-in to the SP:

- SAML2
  - SAML 2.0 Browser Single Logout profile (front- and back-channel)
  - For more complete information about the exact protocol behavior, see the [NativeSPSingleLogoutService](#) topic.
- Local
  - Local removal of a user's session with no IdP involvement

An additional protocol is supported if the relevant extension is loaded:

- ADFS
  - WS-Federation Passive Interoperability Profile (legacy ADFS)
  - For more complete information about the exact protocol behavior, see the [NativeSPSingleLogoutService](#) topic.

Other protocols can be "integrated" with the service-based configuration mechanism by supplying the relevant information via the `<ProtocolProvider>` plugin interface.